In re Application of BEDA et al.
Serial No. 10/693,673

REMARKS

The Office action has been carefully considered. The Office action rejected claims 17 and 60 under 35 U.S.C. §112, second paragraph as being indefinite due to the use of the term "hit-testing." Further, the Office action rejected claims 1-64 under 35 U.S.C. § 101 as being directed to non-statutory subject matter. Further yet, the Office action rejected claims 1-16, 18-59, and 61-64 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Publication 2004/0093604 A1 to Demsey et al. ("Demsey"). Finally, the Office action objected to claim 56 for informality due to a typographical error. Applicants have amended claim 56 to correct the typographical error. Regarding the claim rejections, applicants respectfully disagree.

By present amendment, claims 1, 36, and 56 have been amended for clarification and not in view of the prior art. Applicants submit that the claims as filed were patentable over the prior art of record, and that the amendments herein are for purposes of clarifying the claims and/or for expediting allowance of the claims and not for reasons related to patentability. Reconsideration is respectfully requested.

Applicants thank the Examiner for the interview held (by telephone) on April 25, 2005. During the interview, the Examiner and applicants' attorney discussed the claims with respect to the prior art. The essence of applicants' position is incorporated in the remarks below.

PAGE 16/27 * RCVD AT 5/13/2005 3:12:23 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/7 * DNIS:8729306 * CSID:425 836 8957 * DURATION (mm-ss):07-46

Prior to discussing reasons why applicants believe that the claims in this application are clearly allowable in view of the teachings of the cited and applied references, a brief description of the present invention is presented.

The present invention is directed to a new approach to computer graphics that utilizes a new element object model and a vector graphics markup language for accessing element object models in a manner that allows program code developers to consistently interface with a scene graph data structure to produce graphics. This new system and method includes a scene graph that may have objects that comprise an object model with associated application program interfaces (API). Having objects with APIs allows programmers to accomplish possibly complex composition effects within their applications in a straightforward manner, while leveraging the graphics processing unit in a manner that does not adversely impact normal application performance.

One aspect of the present invention is generally directed to an architecture, referred to as the media integration layer (MIL), that includes an immediate mode graphics application programming interface (API), a screen-partitioning data structure, a set of control level objects, and a markup language. In general, the architecture may allow program code, such as an application or operating system component, to communicate drawing instructions and other information (*e.g.*, image bitmaps) to graphics components in order to render graphical output on the system display. The present invention may provide a number of defined functions and methods, *e.g.*, in the form of APIs to an object model, that enable programs to populate a scene graph with data structures, instruction lists (drawing primitives /

PAGE 17/27 * RCVD AT 5/13/2005 3:12:23 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/7 * DNIS:8729306 * CSID:425 836 8957 * DURATION (mm-ss):07-46

commands), and other graphics-related data. When processed, the scene graph results in graphics being displayed on the screen.

Via the interfaces, the MIL may provide access to a data structure for storing visual information so that applications can take advantage of the graphics capabilities provided by the computer hardware. The interfaces may support an element object model and a vector graphics markup language for using that element object model in a manner that allows program code developers to consistently interface with a scene graph data structure to produce graphics. The data structure may also be used for either directly rendering or for "compiling" the visual information so that it can be provided to a lower level graphics system for fast composition and animation.

Note that the above description is for example and informational purposes only, and should not be used to interpret the claims, which are discussed below.

Rejections under §112

The Office action rejected claims 17 and 60 as being indefinite due to the use of the term "hit-testing." More specifically, the Office action contends that the term, hit-testing, is vague and ambiguous. The Office action goes on further to state that it is not understood as to how one determines the manner in which hit testing should be conducted and that the applicants' disclosure fails to provide an adequate description. Applicants respectfully disagree.

The term hit, as well understood in the art, is a successful retrieval of particular data, a record, or a web site from a cache, database, or computer network. The term, hit, may be contextual and typically refers to the particular

context in which hits are sought, *i.e.*, a relevant web site is a successful hit when searching for web sites, a relevant data record is a successful hit when searching a database of records, *etc.* The process of hit-testing, in turn, is also well understood in the art, as the process of collecting, assembling, and analyzing data associated with a test directed to hits (whatever the context may be).

Thus, in the context of the present invention, it is understood and specifically recited in the specification (see page 188 of the applicants' specification) that hit testing is used to select visuals in a scene. For example, when visuals are called for, (via an API or otherwise), appropriate visuals may be located in a visual tree structure to fulfill such calls. Some high-level scenarios include lasso selection and rubber band selection, keyboard navigation (used to find the next element to switch focus), determining mouse focus in the element tree, selecting overlapping elements with transparency (such as images), "thought bubble" hit testing and selecting a character hit in text.

In general, hit-testing provides consistency across Core, Framework and Controls, and operates by starting from the top of the control tree, and returning a control or set of controls by a point or geometry. A control can define whether it is hit or not with support services including rendered geometry, bounding box, out-of-band geometry (hit region), image opacity or mask, and its own logic. The control can return specific hit-related data on hit (*e.g.*, line, character position, and so forth).

A hit test mechanism may filter hit test results in an efficient manner. Further, the hit test mechanism may provide flexibility for extending to other types

PAGE 19/27 * RCVD AT 5/13/2005 3:12:23 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/7 * DNIS:8729306 * CSID:425 836 8957 * DURATION (mm-ss):07-46

of visuals and resolving down to sub-primitives within the visual, *e.g.*, a Retained3DVisual is one example of this.

Applicants submit that the term, hit-testing, is not vague or ambiguous, whether considered alone or in the context of the applicants' specification. Clearly, there is ample support, both in the general knowledge in the art, as well as within the applicants' own specification, to provide adequate understanding of the use of the term and the concept. Applicants submit that the rejection of claims 17 and 60 under §112, second paragraph is erroneous and request that this rejection be withdrawn.

Rejections under §101

The Office action rejected claims 1-64 as being directed to non-statutory subject matter. Applicants respectfully disagree but have amended claims 1 and 36 to more particularly point out and distinctly claim the subject matter of the invention directed toward a computer-implemented method and a computer system.

Amended claim 1 recites a computer-implemented method that comprises receiving a function call, responding to the function call, and causing a change in a scene graph. Amended claim 36 recites a computer system comprising a scene graph data structure containing data that can be rendered into integrated output that can be viewed and an object model including visual objects having an application program interface and other data that can be contained in the scene graph data structure a data structure and an object model. For at least these

PAGE 20/27 * RCVD AT 5/13/2005 3:12:23 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/7 * DNIS:8729306 * CSID:425 836 8957 * DURATION (mm-ss):07-46

reasons, applicants submit that claims 1-64 are directed to statutory subject matter and that the §101 rejections be withdrawn.

Rejections under §102

Turning to the §102 rejections of the claims, amended claim 1 recites in a computing environment, a computer-implemented method comprising, receiving a function call via an application program interface of an object, the object part of an object model associated with a scene graph, responding to the function call by causing data in the scene graph to be modified.

The Office action rejected claim 1 as being anticipated by Demsey. Specifically, the Office action contends that Demsey teaches a method for arranging computer graphics data for processing into an output, comprising receiving a function call via an interface of an object, the object of an object model associated with a scene graph, responding to the function call by causing data in the scene graph to be modified. The Office action cites FIG. 1 and paragraphs 23, 25, 28, 79, and 91 as support for these rejections. Applicants respectfully disagree.

Demsey is generally directed to a method and system for accessing drawing resources by recognizing two portions of a computer environment, namely a managed code portion and a native code portion. More specifically, communications between the managed code portion and the native code portion involve storing, retrieving, and passing of drawing resources. When a drawing resource is called for within the native code portion of the environment, a value associated with the drawing resource is compared to a value already stored with

PAGE 21/27 * RCVD AT 5/13/2005 3:12:23 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/7 * DNIS:8729306 * CSID:425 836 8957 * DURATION (mm-ss):07-46

the native code. Depending on whether the values are the same or different, the method either retrieves (when the same) or creates (when different) the appropriate drawing resource.

However, Demsey is an example of a system and method for dealing with conventional graphics-related data. Unlike applicants' invention, Demsey does not disclose graphics-related data or drawing resources used in a media integration layer (MIL) with a dedicated application program interface.

In contrast, claim 1 recites receiving a function call via an application program interface. Demsey does not disclose receiving a function call via an application program interface. Further, claim 1 recites that the application program interface is within an object that is part of an object model. Demsey does not disclose utilizing any type of object, let alone an object having an application program interface within an object that is part of an object model. Applicants submit that claim 1 is allowable over the prior art of record for at least these reasons.

Applicants respectfully submit that dependent claims 2-16 and 18-35, by similar analysis, are allowable. Each of these claims depends either directly or indirectly from claim 1 and consequently includes the recitations of independent claim 1. As discussed above, Demsey fails to disclose the recitations of claim 1 and therefore these claims are also allowable over the prior art of record. In addition to the recitations of claim 1 noted above, each of these dependent claims includes additional patentable elements.

PAGE 22/27 * RCVD AT 5/13/2005 3:12:23 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/7 * DNIS:8729306 * CSID:425 836 8957 * DURATION (mm-ss):07-46

For example, claim 2 recites wherein causing data in a scene graph data structure to be modified comprises invoking a function to initialize a new instance of a visual class. Demsey does not disclose a class as commonly used in object-oriented programming. For at least this additional reason, applicants submit that claim 2 is allowable over the prior art of record.

As another example, claim 34 recites wherein causing data in a scene graph data structure to be modified comprises invoking a function to place an object corresponding to audio and/or video data into the scene graph data structure. Again, Demsey does not disclose utilizing objects as commonly understood within the context of object-oriented programming. Thus, Demsey cannot possibly disclose invoking a function to place an object corresponding to audio and/or video data into the scene graph data structure. For at least this additional reason, applicants submit that claim 34 is allowable over the prior art of record.

Turning to the next independent claim, amended claim 36 recites in a computing environment, a computer system comprising a scene graph data structure containing data that can be rendered into integrated output that can be viewed and an object model including visual objects having an application program interface and other data that can be contained in the scene graph data structure.

The Office action rejected claim 36 as being anticipated by Demsey. Specifically, the Office action contends that Demsey teaches a system comprising a scene graph data structure containing data that can be rendered into integrated output that can be viewed and an object model including visual objects and other data that can be contained in the scene graph data structure. The Office action

PAGE 23/27 * RCVD AT 5/13/2005 3:12:23 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/7 * DNIS:8729308 * CSID:425 836 8957 * DURATION (mm-ss):07-46

cites FIG. 1, FIG. 3, FIG. 4, and paragraphs 23, 25, 28, 79, and 91 as support for these rejections. Applicants respectfully disagree.

As discussed above, Demsey is generally directed to a method and system for accessing drawing resources by recognizing two portions of a computer environment, namely a managed code portion and a native code portion. More specifically, communications between the managed code portion and the native code portion involve storing, retrieving, and passing of drawing resources. When a drawing resource is called for at the native code portion of the environment, a value associated with the drawing resource is compared to a value already stored with the native code. Depending on whether the values are the same or different, the method either retrieves (when the same) or creates (when different) the appropriate drawing resource.

However, as also discussed above, Demsey is an example of a system and method for dealing with conventional graphics-related data. Unlike applicants' invention, Demsey does not disclose graphics-related data or drawing resources used in an object model with a dedicated application program interface.

In contrast, claim 36 recites an object model having visual objects that have an application program interface. Demsey does not teach visual objects having an application program interface. Further, claim 36 recites visual objects that can be contained in the scene graph data structure. Demsey does not disclose visual objects that can be contained in the scene graph data structure. Applicants submit that claim 36 is allowable over the prior art of record for at least these reasons.

PAGE 24/27 * RCVD AT 5/13/2005 3:12:23 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/7 * DNIS:8729306 * CSID:425 836 8957 * DURATION (mm-ss):07-46

Applicants respectfully submit that dependent claims 37-59, and 61-64, by similar analysis, are allowable. Each of these claims depends either directly or indirectly from claim 36 and consequently includes the recitations of independent claim 36. As discussed above, Demsey fails to disclose the recitations of claim 36 and therefore these claims are also allowable over the prior art of record. In addition to the recitations of claim 36 noted above, each of these dependent claims includes additional patentable elements.

For at least these additional reasons, applicants submit that all the claims are patentable over the prior art of record. Reconsideration and withdrawal of the rejections in the Office action is respectfully requested and early allowance of this application is earnestly solicited.

PAGE 25/27 * RCVD AT 5/13/2005 3:12:23 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/7 * DNIS:8729306 * CSID:425 836 8957 * DURATION (mm-ss):07-46

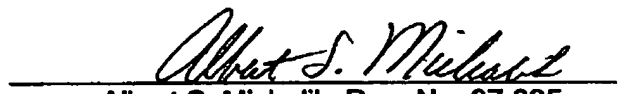
In re Application of BEDA et al.
Serial No. 10/693,673

CONCLUSION

In view of the foregoing remarks, it is respectfully submitted that claims 1-64 are patentable over the prior art of record, and that the application is in good and proper form for allowance. A favorable action on the part of the Examiner is earnestly solicited.

If in the opinion of the Examiner a telephone conference would expedite the prosecution of the subject application, the Examiner is invited to call the undersigned attorney at (425) 836-3030.

Respectfully submitted,



Albert S. Michalik, Reg. No. 37,395
Attorney for Applicants
Law Offices of Albert S. Michalik, PLLC
704 - 228th Avenue NE, Suite 193
Sammamish, WA 98074
(425) 836-3030
(425) 836-8957 (facsimile)

PAGE 26/27 * RCVD AT 5/13/2005 3:12:23 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/7 * DNIS:8729306 * CSID:425 836 8957 * DURATION (mm-ss):07-46

In re Application of BEDA et al.
Serial No. 10/693,673

CERTIFICATE OF FACSIMILE TRANSMISSION

I hereby certify that this Response, along with transmittal and facsimile cover sheet, are being transmitted by facsimile to the United States Patent and Trademark Office in accordance with 37 C.F.R. 1.6(d) on the date shown below:

Date: May 13, 2005

Albert S. Michalik

3471 Amendment

PAGE 27/27 * RCVD AT 5/13/2005 3:12:23 PM [Eastern Daylight Time] * SVR:USPTO-EFXRP-1/7 * DNIS:8729306 * CSID:425 836 8957 * DURATION (mm-ss):07-46